The Annealing Sparse Bayesian Learning Algorithm

Benyuan Liu, Hongqi Fan, *Member, IEEE,* Zaiqi Lu, Qiang Fu

EDICS Category: SAS-MALN

***Abstract*—In this paper we propose a two-level hierarchical Bayesian model and an annealing schedule to re-enable the noise variance learning capability of the fast marginalized Sparse Bayesian Learning Algorithms. The performance such as NMSE and F-measure can be improved due to the annealing technique. This algorithm tends to produce the most sparse solution under moderate SNR scenarios and can outperform most concurrent SBL algorithms while pertains small computational load.**

***Index Terms*—bayesian methods, compressive sensing, sparse bayesian learning, fast marginalized, annealing**

I. Introduction

The Sparse Bayesian Learning(SBL) algorithms [1]–[3] recast the solution to compressive sensing [4]–[6] in a probabilistic way. One of the advantages of SBL over traditional convex optimization algorithms [7], [8] is it's free of choosing regularized penalty parameters. The noise variance σ^2 along with other hyper-parameters can be automatically learned during the iterative procedure. Such typical SBL algorithms include EMSBL [1]–[3], [9] and TMSBL [10]. In order to reduce the computational time of SBL algorithms, fast marginalized methods [11], [12] have been utilized, but those algorithms (BCS [13] and FLSBL [14]) require the user to specify a proper noise variance and are void of automatic σ^2 learning capability. In this paper we propose a two-level hierarchical Bayesian model and a novel annealing technique to re-enable the noise learning capability of fast marginalized algorithms. The proposed algorithm is fast and outperforms most concurrent SBL algorithms in terms of NMSE and the number of relevant basis.

II. Bayesian Hierarchical Model

The Single Measurement Vector(SMV) form of sparse signal reconstruction problem is:

$$\mathbf{y} = \Phi\mathbf{w} + \mathbf{n}, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^{M\times 1}$ is the measurement vector, Φ is $M \times N$ measurement matrix with $M \ll N$, $\mathbf{w} \in \mathbb{R}^{N\times 1}$ is the signal to be recovered, and $\mathbf{n}$ is an i.i.d. Gaussian with zero mean and variance equal to β^{-1}.

In Bayesian modeling, each unknown quantity is modeled as a stochastic variable. The two-level hierarchical Bayesian model is constructed as:

$$p(\mathbf{y}|\mathbf{w},\beta) = \mathcal{N}(\mathbf{y}|\Phi\mathbf{w},\beta^{-1}) \tag{2}$$

$$p(w_i|\gamma_i) = \mathcal{N}(w_i|0,\gamma_i B) \tag{3}$$

where (2) is called observation model, and (3) is signal model. Each coefficient w_i is modeled as a Gaussian Process with variance equal to $\gamma_i B$, in which B is a scalar and will be investigated in detail in section IV. We will use the improper [1] hyper-prior for parameters γ_i, β and B.

The probability of hyper-parameters conditioned on observed data $\mathbf{y}$ is

$$p(\mathbf{w},\boldsymbol{\gamma},B,\beta|\mathbf{y}) = p(\mathbf{w}|\mathbf{y},\boldsymbol{\gamma},B,\beta)p(\boldsymbol{\gamma},B,\beta|y) \tag{4}$$

These parameters can be estimated using a Type II maximum likelihood procedure [1]:

$$\hat{\boldsymbol{\gamma}},\hat{B},\hat{\beta} = \arg\max_{\boldsymbol{\gamma},B,\beta} p(\mathbf{y}|\boldsymbol{\gamma},B,\beta) \tag{5}$$

using Bayes' rule we have:

$$p(\mathbf{w}|\mathbf{y},\boldsymbol{\gamma},B,\beta)p(\mathbf{y}|\boldsymbol{\gamma},B,\beta) = p(\mathbf{y}|\mathbf{w},\beta)p(\mathbf{w}|\boldsymbol{\gamma},B) \tag{6}$$

where the right hand side of (6) is given in (2) and (3), we can solve (6) using Gaussian Identities:

$$p(\mathbf{w}|\mathbf{y},\boldsymbol{\gamma},B,\beta) = \mathcal{N}(\mathbf{w}|\mu,\Sigma) \tag{7}$$

$$p(\mathbf{y}|\boldsymbol{\gamma},B,\beta) = \mathcal{N}(\mathbf{y}|0,C) \tag{8}$$

By taking the partial derivatives of $\mathcal{L} = \log p(\mathbf{y}|\boldsymbol{\gamma},\beta,B)$ with respect to γ_i, B, β and setting them equal to 0, the update rules of the those hyper-parameters can be obtained:

$$\gamma_i = \frac{1}{B}(\mu_i^2 + \Sigma_{ii}) \tag{9}$$

$$B = \frac{1}{N}(\mathrm{Tr}[\Sigma\Lambda] + \mu^T\Lambda\mu) \tag{10}$$

$$\beta = \frac{N}{\|\mathbf{y} - \Phi\mu\|^2 + \mathrm{Tr}[\Sigma\Phi^T\Phi]} \tag{11}$$

where $\Lambda = \mathrm{diag}(1/\gamma_i)$, μ_i is the ith element of μ and Σ_{ii} is the ith diagonal element of matrix Σ. The derivation is similar to TMSBL [10] except that we have explicitly model B as a scalar.

III. Fast Marginalized Implementations

We rewrite the covariance of $p(\mathbf{y}|\boldsymbol{\gamma},\beta,B)$ as:

$$\mathbf{C} = \beta^{-1}\mathbf{I} + \sum_i B\gamma_i\phi_i\phi_i^T \tag{12}$$

$$= \beta^{-1}\mathbf{I} + \sum_{j\neq i} B\gamma_j\phi_j\phi_j^T + B\gamma_i\phi_i\phi_i^T \tag{13}$$

$$= \mathbf{C}_{-i} + B\gamma_i\phi_i\phi_i^T \tag{14}$$

where ϕ_i is the ith column (basis) of Φ and $\mathbf{C}_{-i}$ denotes that the contribution of ith basis is excluded from $\mathbf{C}$. This equation has an additional weight B compared with BCS [13] and FLSBL [14]. The update rule for γ_i is

$$\gamma_i = \frac{1}{B}\frac{q_i^2 - s_i}{s_i^2} \tag{15}$$

Benyuan Liu is with the Department of Electrical and Communication Engineering, University of Defense Technology, Changsha, Hunan, 410074, China. e-mail: liubenyuan@gmail.com

Manuscript received September 1, 2012.

where s_i and q_i is defined as $s_i = \phi_i^T \mathbf{C}_{-i}^{-1} \phi_i$, $q_i = \phi_i^T \mathbf{C}_{-i}^{-1} \mathbf{y}$. The process of **Add**, **Delete**, and **Re-Estimate** is identical to Tipping [11], [12]. For a given basis i, the change of $\mathcal{L}$ under **Add**, **Delete**, and **Re-Estimate** is denoted as $\Delta\mathcal{L}(\gamma_i) = \mathcal{L}(\tilde{\gamma}_i) - \mathcal{L}(\gamma_i)$, where $\tilde{\gamma}_i$ is the updated value for γ_i. By calculating $\Delta\mathcal{L}(\gamma_i), \forall i$, the one which maximize the change in $\mathcal{L}$ is selected to be updated to boost the convergence speed. The change of log-likelihood is also used to test the convergence of the algorithm.

We should note that each time when β or B is altered, the whole quantities of fast marginalized algorithms such as s_i,q_i,μ and Σ must be re-calculated, we denote it as the process of **Update**(see [12]).

IV. The Annealing SBL

The role of B is analyzed by exploring the structure of $\mathbf{C}$:

$$C = \beta^{-1}\mathbf{I} + B\Phi\Lambda^{-1}\Phi^T \tag{16}$$

As Wipf [3] and Zhang [10] pointed out, given $\Phi = [\Phi', \mathbf{I}]$ and $\mathbf{I}$ is $M \times M$ identity matrix, the above equation could be rewritten as:

$$\begin{aligned} C &= \beta^{-1}\mathbf{I} + B\Phi\Lambda^{-1}\Phi^T \qquad (17)\\ &= \beta^{-1}\mathbf{I} + B\Phi'\Lambda'^{-1}\Phi'^T + B\mathrm{diag}(\gamma_{N-M+1}, \cdots, \gamma_N) \qquad (18)\end{aligned}$$

With $B = 1$, a nonzero value of β and M nonzero values of $\gamma_{N-M+1}, \cdots, \gamma_N$ make identical contribution to the covariance matrix $\mathbf{C}$, thus β and $\boldsymbol{\gamma}$ are not identifiable which leading to degrade performance [10]. This is especially true in BCS [13] and FLSBL [14] due to the constructive and reconstruction manner of the algorithm. We also observe that when B takes a small value, the portion of learning error of $\boldsymbol{\gamma}$ contributing to the overall $\mathbf{C}$ can be minimized and β dominates the covariance matrix. During the iterative learning process, the accuracy of γ_i is improved which suggests that the restriction on $\boldsymbol{\gamma}$ in the covariance matrix $\mathbf{C}$ could be released by increasing the value of B.

Inspired by the continuation strategy by Hale [15] and the simulated annealing methods, *it is possible to select an arbitrary large noise variance as the initial value for σ^2 and adopt an increasing sequence of B to obtain the solution of $\hat{\mathbf{w}}$ and an estimate of σ^2*, which is illustrated in Figure 1.

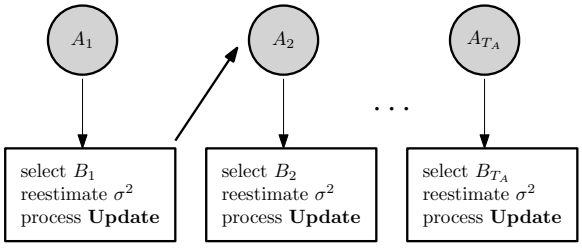


Fig. 1. The Annealing SBL Scheme. The annealing steps is denoted as T_A. At each step i, the annealing update criterion is A_i, a new value of B_i is selected and β is calculated. All the quantities of the algorithm such as μ, Σ, s_i and q_i should also be **Updated**.

In what follows, I will give explicit formulas of some key ingredients of the annealing schedule: (1) the annealing update criterion and (2) the annealing step size.

A. The Annealing Update Criteria

In Ji [13] and Babacan [14], the program converges when the change between consecutive $\Delta\mathcal{L}$ is less than the change between current and first $\Delta\mathcal{L}(1)$, which is

$$\frac{|\Delta\mathcal{L}(k) - \Delta\mathcal{L}(k-1)|}{|\Delta\mathcal{L}(k) - \Delta\mathcal{L}(1)|} \leq \eta. \tag{19}$$

Given an annealing step size T_A, the annealing criterion is

$$\eta_A^{T_A} = \eta \tag{20}$$

$$\frac{|\Delta\mathcal{L}(k_A) - \Delta\mathcal{L}(k_A - 1)|}{|\Delta\mathcal{L}(k_A) - \Delta\mathcal{L}(1_A)|} \leq \eta_A. \tag{21}$$

In the above equation, k_A is the iterative number at an annealing temperature A. We can see that each annealing step is made to decrease the change of log-likelihood to a fraction of $\Delta\mathcal{L}(1)$, the overall exit criterion is made to be the same with the BCS [13] and FLSBL [14] algorithm. We also find that (20) is too slow for B_i taking 1. We thus modify (20) with a scaling factor:

$$\eta_A^{\alpha T_A} = \eta, \tag{22}$$

and $\alpha = 2$ is used in our experiments.

B. The Annealing Steps T_A

The increasing sequence of $\mathbf{B}$ is uniformly divided in the interval $[0.1, 1]$ with T_A steps. This parameter is analyzed in detail in the next section. Each time when the annealing criterion is met, we select the next B_i in the sequence $\mathbf{B}$. The σ^2 can be automatically updated in the Fast Marginalized algorithm and the initial value of σ_i^2 can be selected arbitrary. For convenience we simply let $\sigma_i^2 = \|\mathbf{y}\|^2$.

C. The ASBL Algorithm

The Annealing SBL algorithm is given in Fig. 2. The proposed method is called **ASBL** in the remaining of this paper.

```
procedure ASBL(Φ, y, η, σ_i^2, T_A)
    while Global Convergence is not met do
        if Annealing Criterion is met then
            select B_i from B.
            update β using eq(11).
            process Update.
        end if
        process BCS routine.
    end while
end procedure
```

Fig. 2. The ASBL Algorithm

V. Experiments

A signal $\mathbf{w}$ of length N is generated with T nonzero weights random located, the amplitude of each nonzero weights is sampled from uniform ± 1 random spikes. The measurement matrix $\Phi \in \mathbb{R}^{M \times N}$ is a uniform spherical ensemble, with each columns ϕ_i uniformly distributed on the sphere $\mathbb{R}^N$. $\mathbf{n}$ is an

i.i.d. Gaussian variable with variance σ_n^2, the standard deviation of noiseless measurements $\mathbf{y}_s = \Phi\mathbf{w}$ is σ_y. The signal-to-noise ratio (SNR) is defined as $\text{SNR} = 20\log_{10}\sigma_y/\sigma_n$.

We compare the proposed method with the algorithms EMSBL [1], [9], TMSBL [10], BCS [13] and FLSBL [14]. In moderate SNR scenarios, the F-measure of Support Recovery(F-index) [10] was used as a performance index, defined by $F = |\Theta_c|/|\Theta_t|$ and $\Theta_c = \Theta_e \cap \Theta_t$, where Θ_t was the locations of true signal $\mathbf{w}$ and Θ_e was the maximum T locations of the estimated signal $\hat{\mathbf{w}}$. We also calculate the normalized Mean Square Error(NMSE), defined by $\|\hat{\mathbf{w}}-\mathbf{w}\|_2^2/\|\mathbf{w}\|_2^2$, as well as the CPU time and the number of relevant basis, which is denoted as N_B.

For simplicity, in the experiments thereafter we fix $M = 100$, $N = 512$ and vary T and SNR to test these algorithms under different sparsity and noise levels. A similar phase transition is used to illustrate how the sparsity level (defined by $\rho = T/M$) and noise level affect the *success* of the algorithm, where a *success* is defined when the average of F-index exceed 0.9. A point above the phase plot indicates a failure while below the curve the success is 1. We vary the SNR from 5dB to 25dB with 5dB step size and run each experiment for 100 iterations.

A. The choice of T_A

The phase transitions of ASBL with different T_A values is plotted in Figure 3. The values $T_A = 8, 10$ have similar

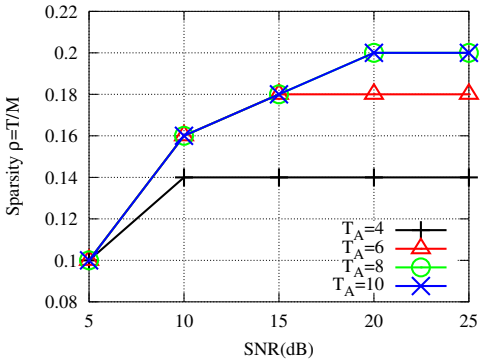


Fig. 3. The Choice of T_A. The phase transition with different sparsity levels and SNR is plotted. We set the annealing steps of ASBL to $T_A = 4, 6, 8, 10$ and the convergence criterion $\eta = 10^{-4}$.

performance, while $T_A = 8$ takes small annealing steps and computational load. We will choose $T_A = 8$ as the default parameter of ASBL in the remaining experiments.

B. The Phase Transition

ASBL is inherently BCS with additional B annealing schedules and σ^2 learning capabilities. In this experiment we will plot the phase transition with respect to different sparsity and SNR levels for ASBL, BCS, FLSBL, EMSBL and TMSBL. The true noise variance is selected as the initial σ_i^2 value for BCS and FLSBL. For EMSBL and TMSBL, the automatic σ^2 learning capability is toggled on. The result is shown in Figure 4. The performance of ASBL, BCS and FLSBL is inferior to

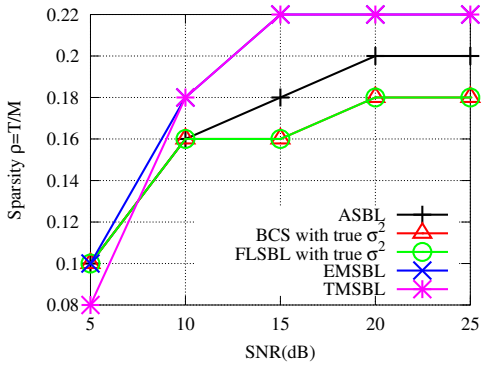


Fig. 4. The phase transitions of ASBL, BCS, FLSBL, EMSBL and TMSBL. The phase transition is plotted with respect to different sparsity and SNR levels. Each point on the phase curve corresponds to the average of F-index larger than or equal 0.9.

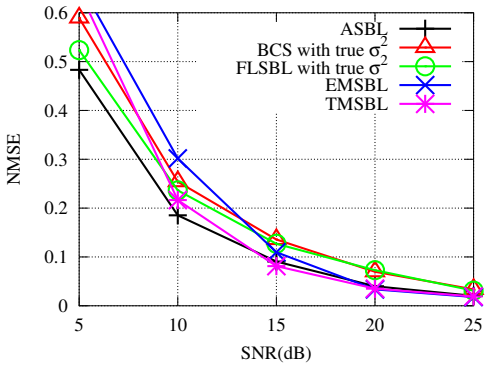
EMSBL and TMSBL when SNR $\geq$ 10dB and sparsity level $\rho > 0.2$, this is an open issue largely due to the constructive and reconstruction nature of the fast marginalized method, which will be explored in our next paper. The ASBL algorithm has better performance than BCS and FLSBL, this property is attained without even a prior knowledge of the true noise variance. The advantage of introducing additional annealing steps will be analyzed in detail in the next experiment.

C. The performance comparison of different SBL algorithms on 1D data

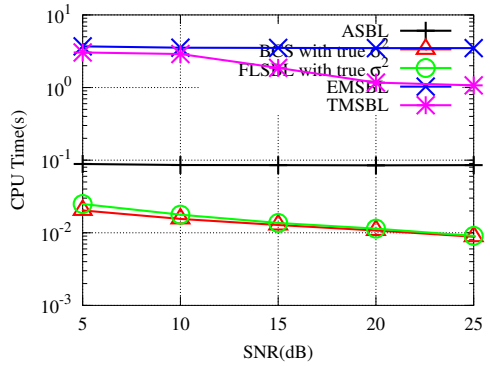
In this experiment we fix $M = 100$, $T = 10$, SNR $= 10$dB and compare different SBL algorithms in term of NMSE, CPU time and the number of relevant basis N_B. The simulation results is plotted in Figure 5. It is interesting that the ASBL algorithm seems to attain the lower bound in terms of NMSE among those algorithms when SNR < 15dB. The average time of TMSBL and EMSBL is 3s and 5s respectively, while ASBL takes only a little longer than BCS and FLSBL. The number of relevant basis N_B of ASBL is the smallest among all the algorithms, which means that the proposed method produces the most sparse solution under moderate SNR scenarios. This is very impressive given its superior performance in NMSE and CPU time.

D. The σ^2 learning capabilities of SBL algorithms

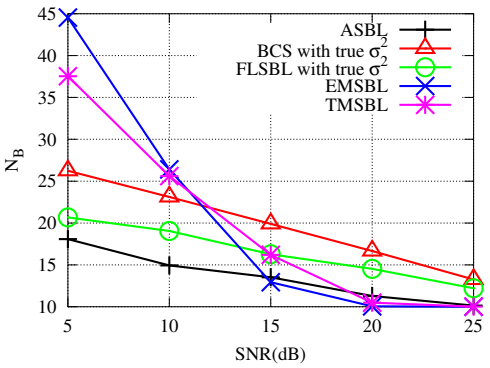
The estimated of σ^2 of ASBL, as well as true noise variance and the estimated σ^2 of different SBL algorithms is plotted in Figure 6. We can see that the BCS, FLSBL, EMSBL tend to under-estimate the noise variance during the learning process, while TMSBL with the advanced σ^2 learning option toggled on tend to over-estimate the noise level. The BCS and FLSBL have the same slope as ASBL. Among those algorithms, the ASBL showed the best σ^2 learning performance, which means that with an arbitrary large initial noise variance ($\sigma_i^2 = \|\mathbf{y}\|_2^2$) as the starting point, the proposed annealing method gradually seeks the balance between data misfit and regularized penalty, produces an accurate estimation of σ^2 when the annealing procedure stops.



(a) NMSE



(b) CPU Time(s)



(c) N_B

Fig. 5. NMSE, CPU time and N_B versus SNR with different SBL algorithms. The automatic σ^2 learning option is toggled on for EMSBL and TMSBL. The initial value of σ^2 for BCS and FLSBL is set to the true noise variance. The convergence criterion are $\eta = 10^{-4}$ for ASBL, BCS and FLSBL, $\eta = 10^{-8}$ for EMSBL and TMSBL.

VI. Conclusion

In this paper we propose an annealing SBL (**ASBL**) algorithm and an implementation using Fast Marginalized method. The ASBL algorithm is free of user tuned parameters and can automatically update the noise variance σ^2 to lock to the optimum performance during the learning process. The ASBL tends to produce the most sparse solution under moderate SNR (SNR $<$ 15dB) and its performance is superior to TMSBL, EMSBL, BCS and FLSBL in terms of NMSE. These properties are very attractive for signal reconstruction in noisy measurements. The proposed method is based on fast marginalized implementation and its CPU time is far more less than EMSBL and TMSBL which will win ASBL a broad area of applications.

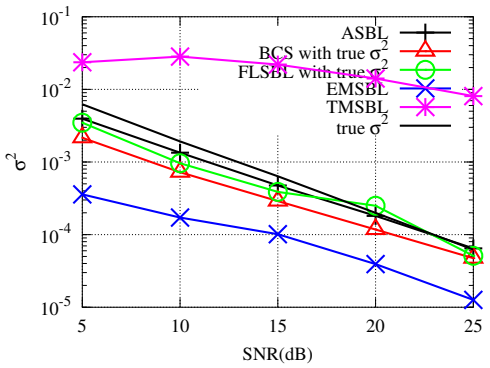


Fig. 6. The estimated noise variance versus SNR with different SBL algorithms. The EMSBL and TMSBL have automatic σ^2 learning capability. For BCS and FLSBL, the σ^2 is updated when the convergence criterion has met.

Acknowledgement

The author thanks Dr. Zhilin Zhang of UCSD for his simulation code and discussions.

References

[1] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[2] D. Wipf and B. Rao, "Sparse bayesian learning for basis selection," *Signal Processing, IEEE Transactions on*, vol. 52, no. 8, pp. 2153 – 2164, aug. 2004.

[3] ——, "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *Signal Processing, IEEE Transactions on*, vol. 55, no. 7, pp. 3704 –3716, july 2007.

[4] E. Candes and M. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21 –30, march 2008.

[5] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 489 – 509, feb. 2006.

[6] R. Baraniuk, "Compressive sensing [lecture notes]," *Signal Processing Magazine, IEEE*, vol. 24, no. 4, pp. 118 –121, july 2007.

[7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20(1), pp. 33–61, 1998.

[8] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using focuss: a re–weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, pp. 600–616, 1997.

[9] M. E. Tipping, "Bayesian inference : An introduction to principles and practice in machine learning," in *Advanced Lectures on Machine Learning*, O. Bousquet, U. von Luxburg, and G. Ratsch, Eds. Springer, 2004, pp. 41–62.

[10] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 1–15, 2011.

[11] A. C. Faul and M. E. Tipping, "Analysis of sparse bayesian learning," *Neural Inform. Process. Syst.*, vol. 14, pp. 383–389, 2002.

[12] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse bayesian models," in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, C. M. Bishop and B. J. Frey, Eds., Key West, FL, 2003, pp. 3–6.

[13] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, pp. 2346–2356, 2008.

[14] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using laplace priors," *IEEE Transactions on Signal Processing*, vol. 19, pp. 53–63, 2010.

[15] E. T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing," CAAM Technical Report TR07-07, Rice University, Tech. Rep., 2007.